Computational Thinking and Making: Computational Making

invited talk

Connecting Technologies and Didactics - The IDEA Project Experience

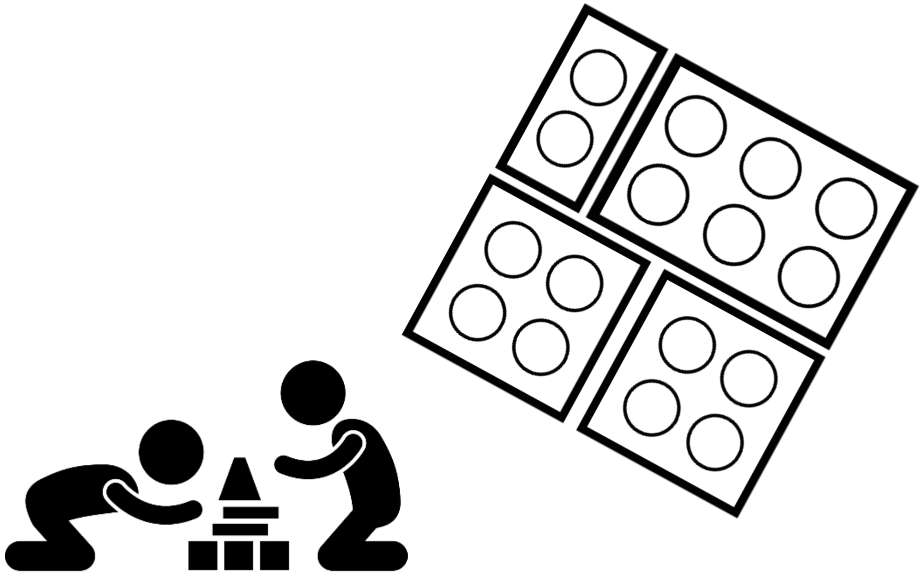
Center for Advanced Studies Research and Development - CRS4

Thotel, Cagliari, February 10th to 11th 2020

Daniel K. Schneider
tecfa
Associate professor of educational technology
Faculty of Psychology and Educational Sciences
University of Geneva
http://tecfa.unige.ch/DKS
http://edutechwiki.unige.ch/

https://tecfa.unige.ch/tecfa/talks/schneide/crs4-2020/

https://tecfa.unige.ch/tecfa/talks/schneide/crs4-2020/



1. CONTEXT, DEFINITIONS AND OBJECTIVES

“Making” = Digital design and fabrication

Some end-user making technology:

Computerized Embroidery:

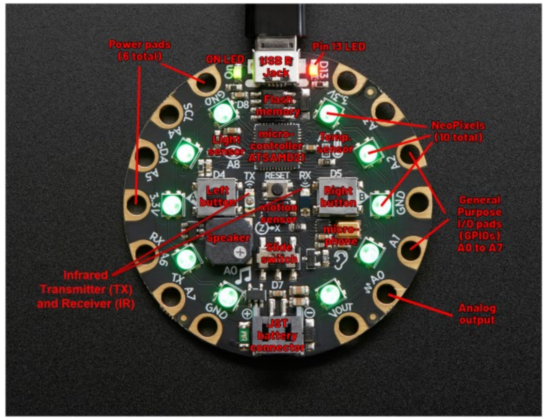
- Medium or expensive hardware
- Very expensive or free software
- Environment friendly
- noisy

Laser cutting:

- Expensive hardware
- Free or cheap software
- Fast
- Very noisy, smelly,

Electronic boards

- Cheap hardware
- Free software
- require programming



Vinyl cutting

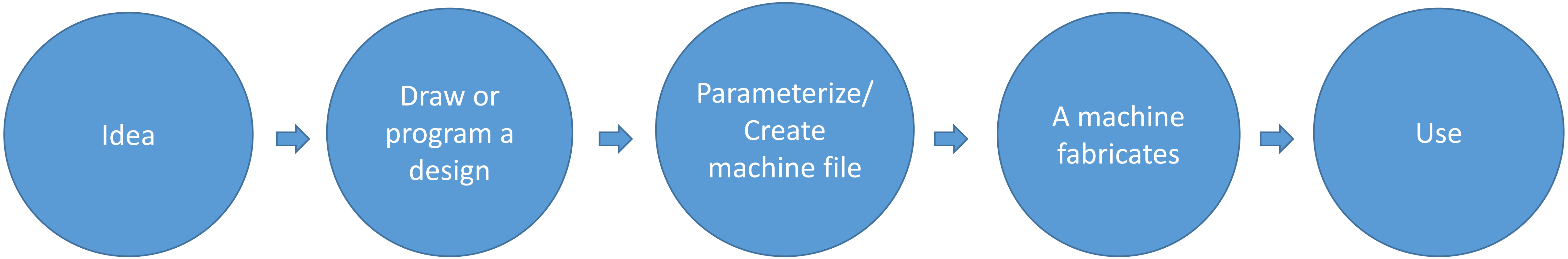
- Very cheap hardware
- Rather cheap software
- Fast

3D printing

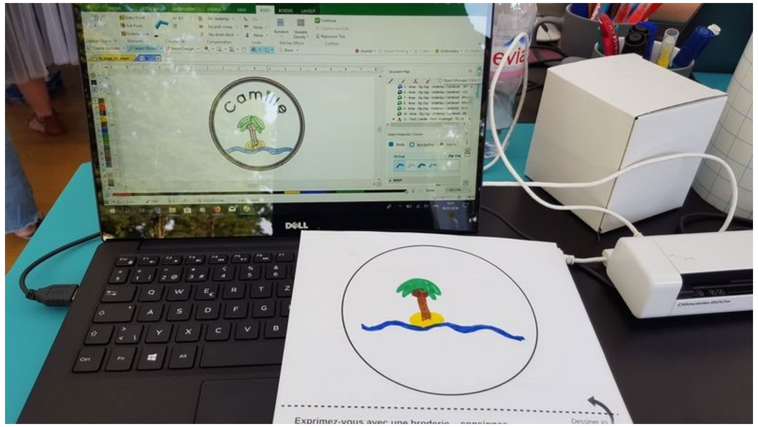
- Cheap hardware
- Free software
- Very slow
- Smelly

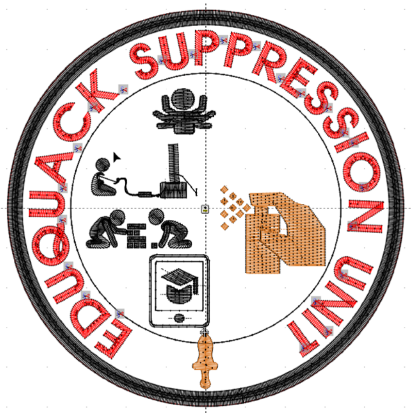
Gershenfeld (2005) «How to make almost anything » / Bowyer (2007) RepRap

Workflow in making (simplified)

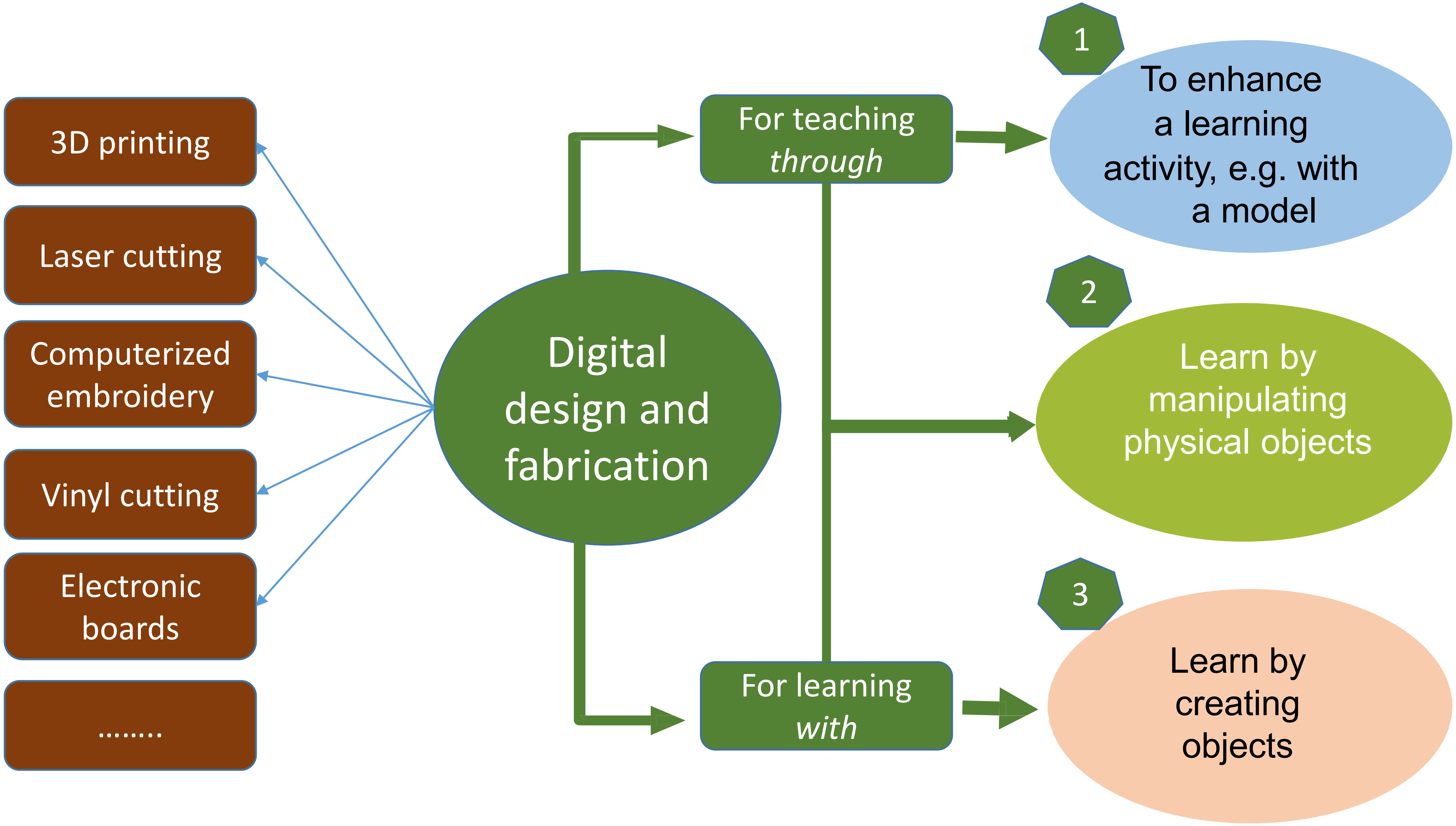


Idea → Draw or program a design → Parameterize/ Create machine file → A machine fabricates → Use





Making in education (1)



3D printing

Laser cutting

Computerized embroidery

Vinyl cutting

Electronic boards

……..

Digital design and fabrication

For teaching *through*

For learning *with*

1

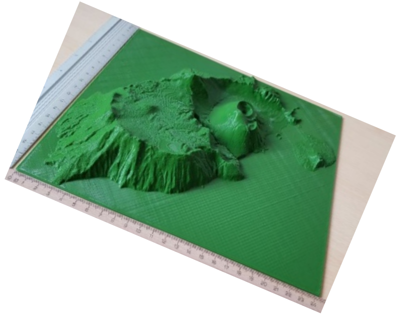
To enhance a learning activity, e.g. with a model

2

Learn by manipulating physical objects

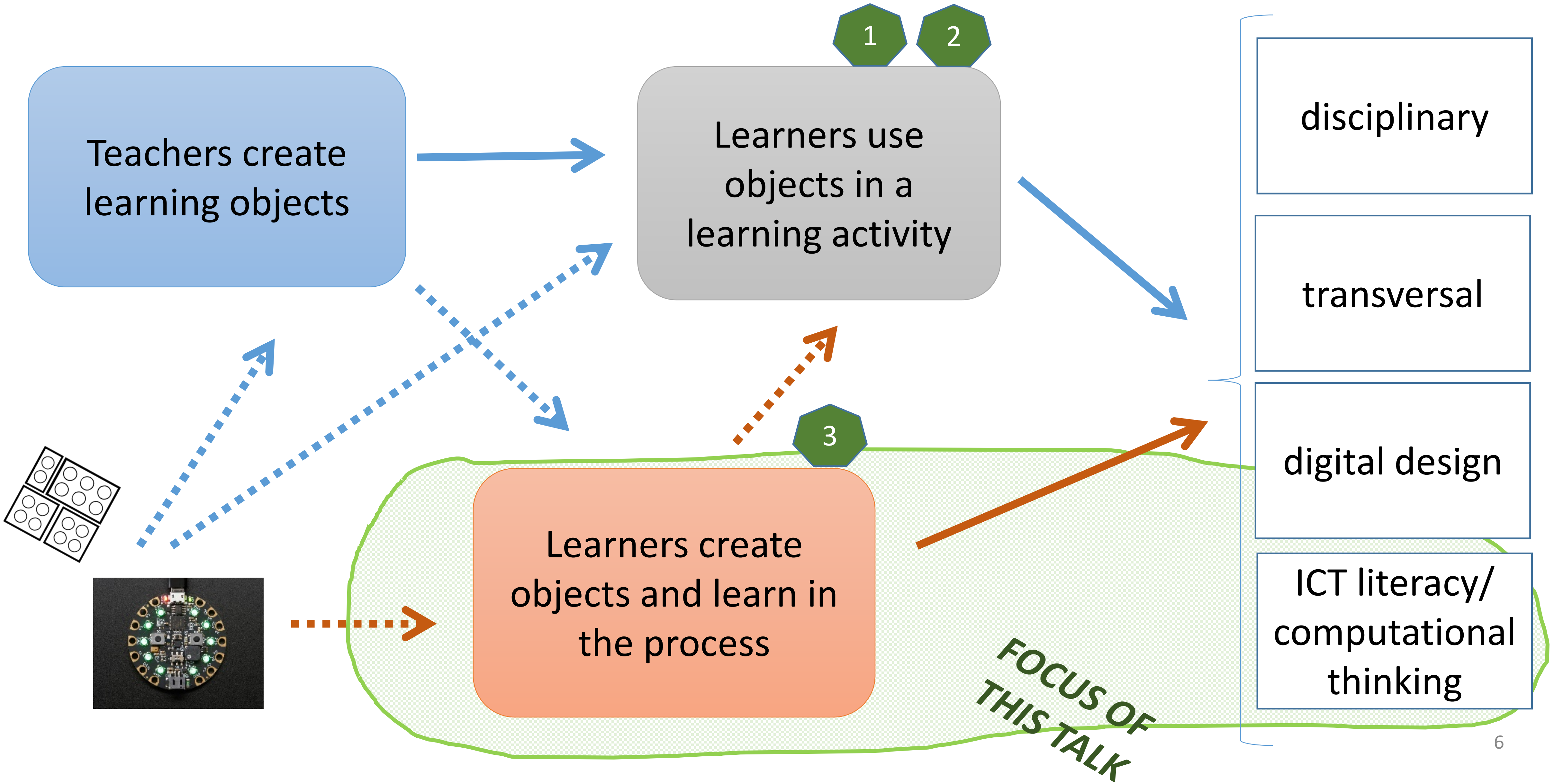
3

Learn by creating objects



Making in education (2)

Skills



Teachers create learning objects

Learners use objects in a learning activity

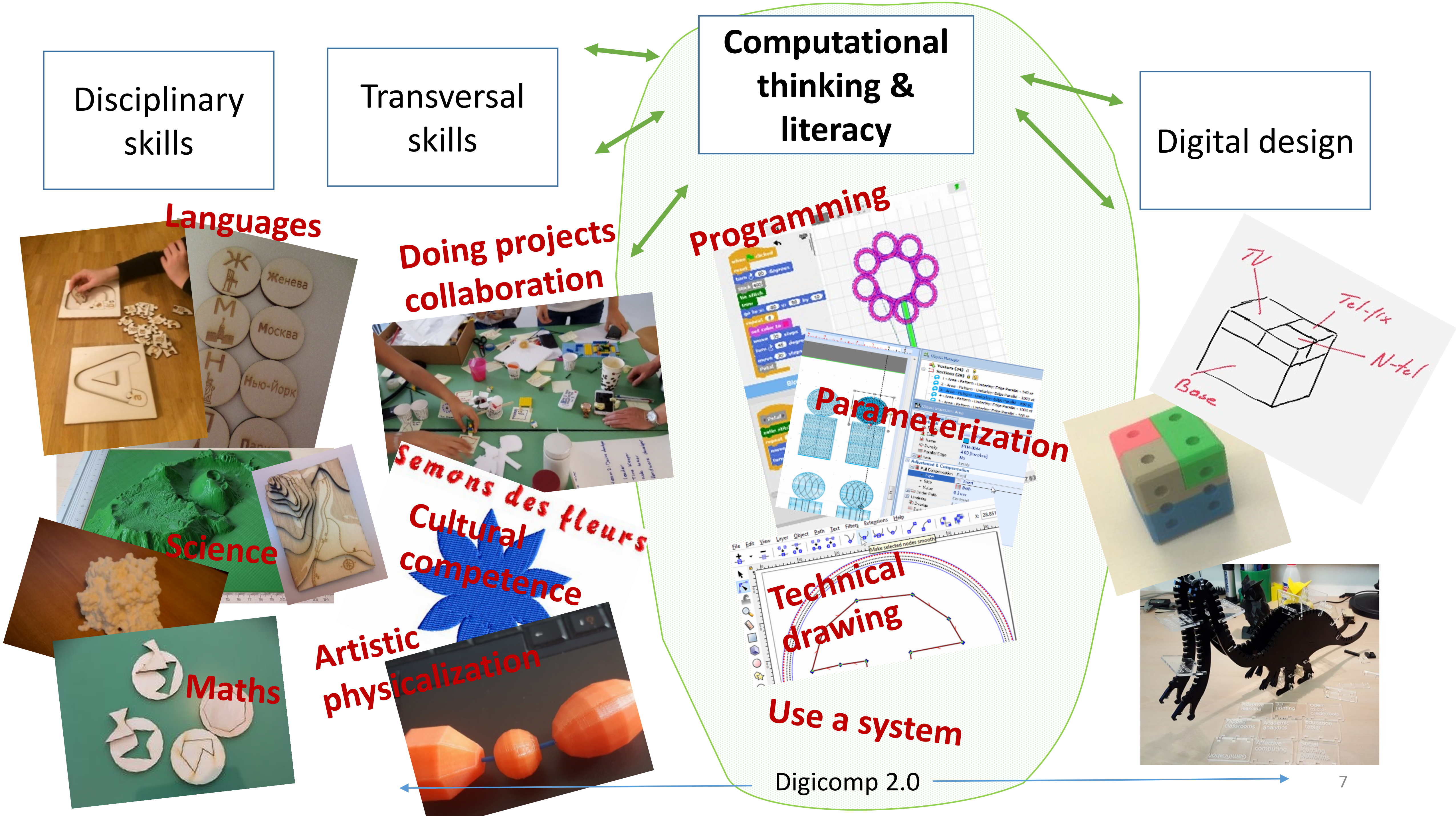
Learners create objects and learn in the process

disciplinary

transversal

digital design

ICT literacy/ computational thinking



Disciplinary skills

Transversal skills

Computational thinking & literacy

Digital design

Languages

Doing projects collaboration

Programming

Parameterization

Technical drawing

Use a system

Science

Semons des fleurs

Cultural competence

Artistic physicalization

Maths

Digicomp 2.0

Computational thinking

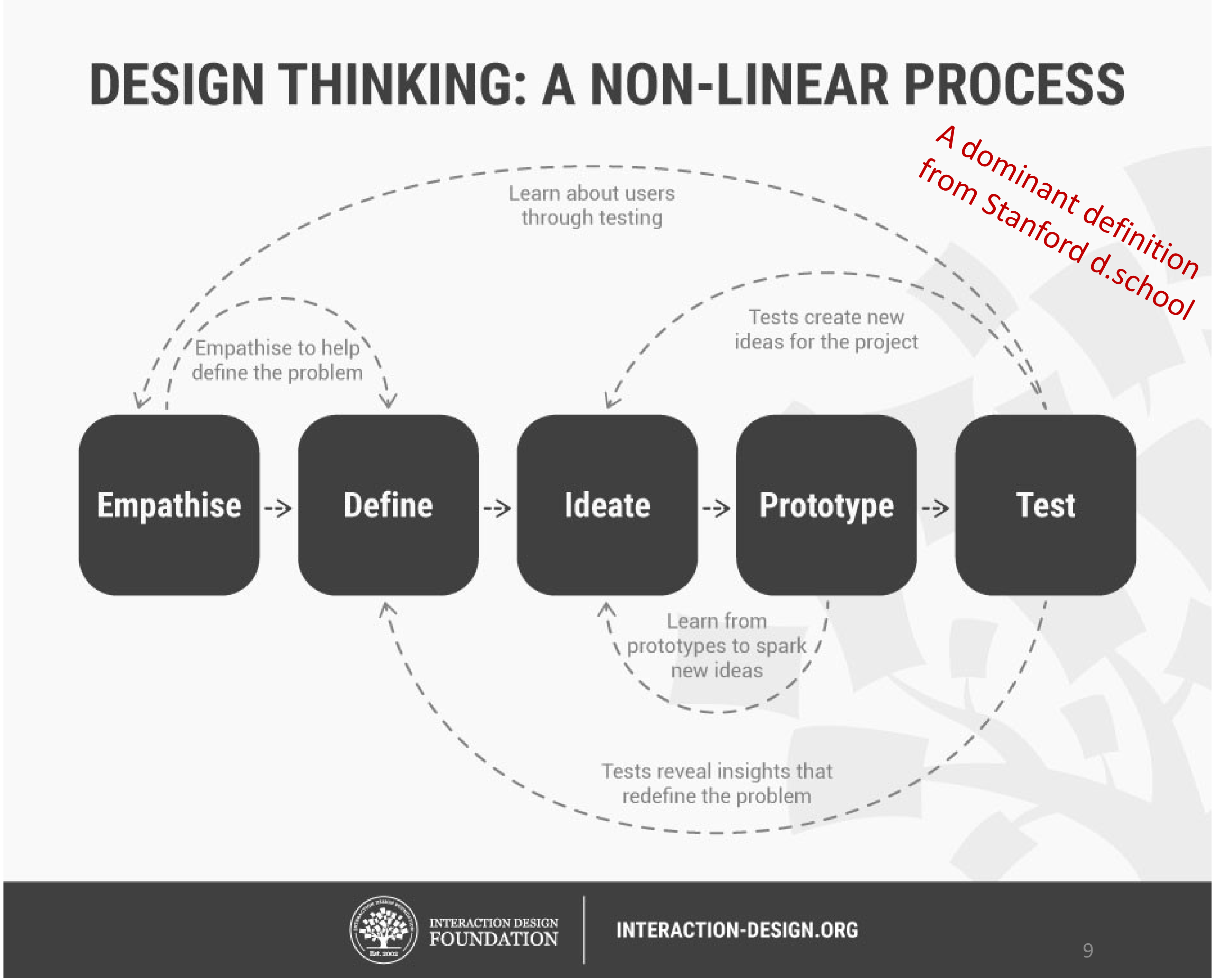
a set of knowledge, attitudes and skills that facilitate problem solving based on principles from computer science.

Computational thinking competence:
(1) formulate a broader issue
(2) solve a problem / create a solution using information and communication technologies*
(3) share solutions

*using ICT:

1. decompose, abstract, define algorithm, identify forms, ……
2. find and install software, draw, manipulate images, parameterize…..

Both **Making** and **Computational Thinking** are related to **Design Thinking** (some kind of user-centered design)



DESIGN THINKING: A NON-LINEAR PROCESS

Empathise --> Define --> Ideate --> Prototype --> Test

Learn about users through testing

Empathise to help define the problem

Tests create new ideas for the project

Learn from prototypes to spark new ideas

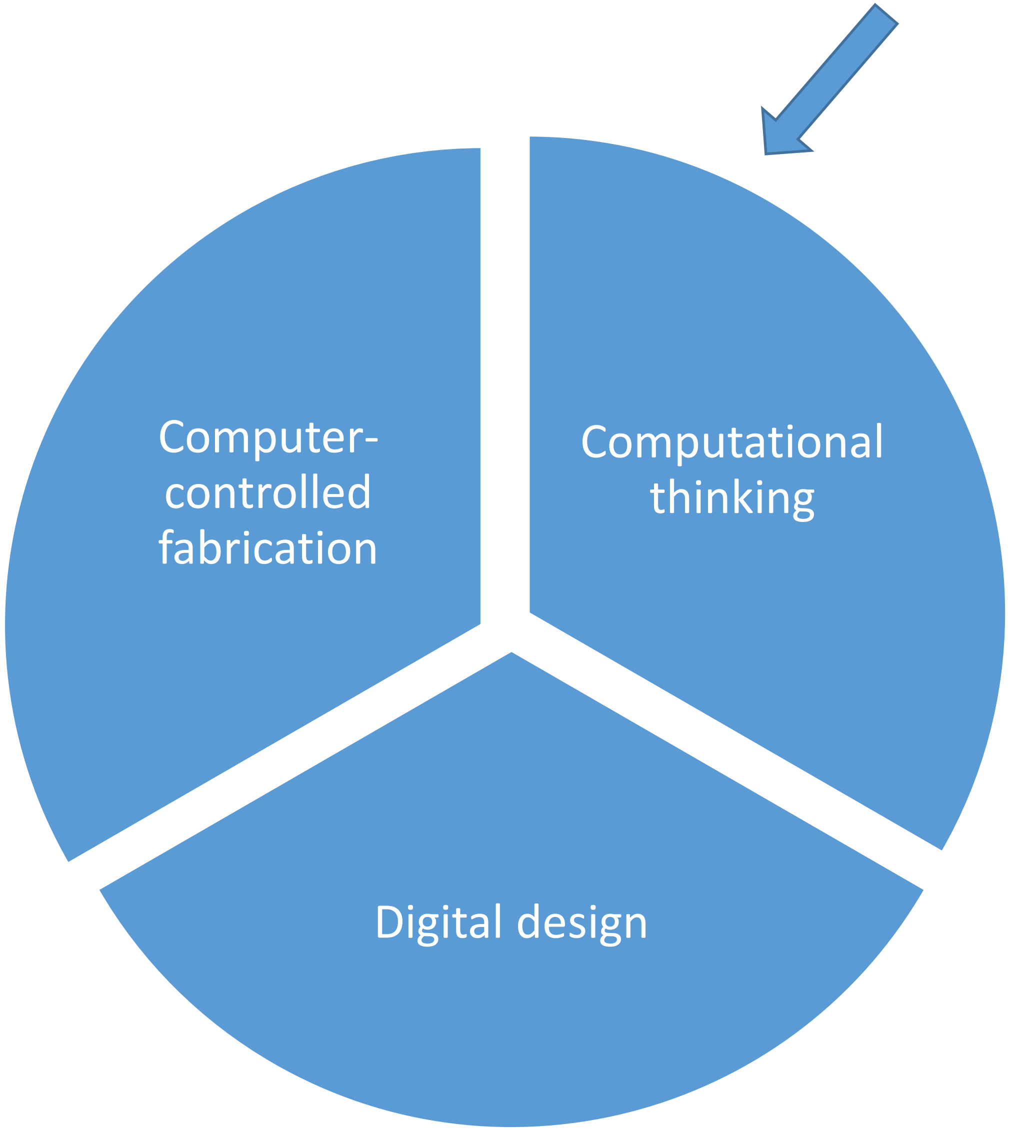
Tests reveal insights that redefine the problem

INTERACTION DESIGN FOUNDATION

INTERACTION-DESIGN.ORG

A dominant definition from Stanford d.school

Computational making



Computational making

- Is essentially computational design,
- affords high levels of precision,
- automation of repetitive tasks,
- adjust values while maintaining the constraints of the original model (parameterization).
- Algorithms produce unique & unexpected designs.

Computational design = *using programming to create and modify form, structure, and ornamentation* (Jacobs and Buechley, 2013).

Questions ?

Can we teach computational thinking
(and other ICT skills) through making ?

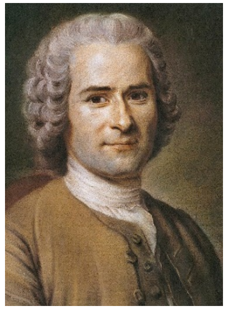
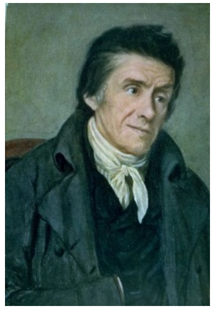
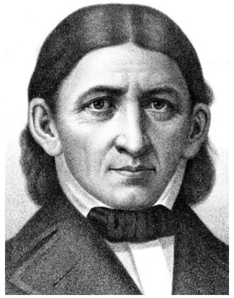
What environments and activities work ?

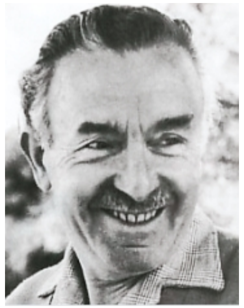
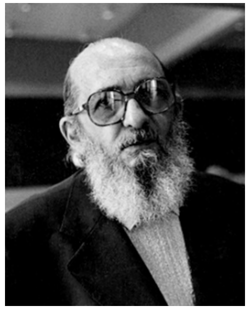
Is it effective (and efficient) ?

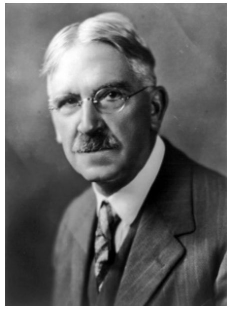
https://tecfa.unige.ch/tecfa/talks/schneide/crs4-2020/

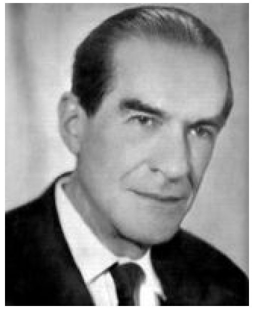
Following the review of Zuckerman (2006) on constructivst learning objects

2. THEORETICAL FOUNDATIONS FOR LEARNING WITH THINGS

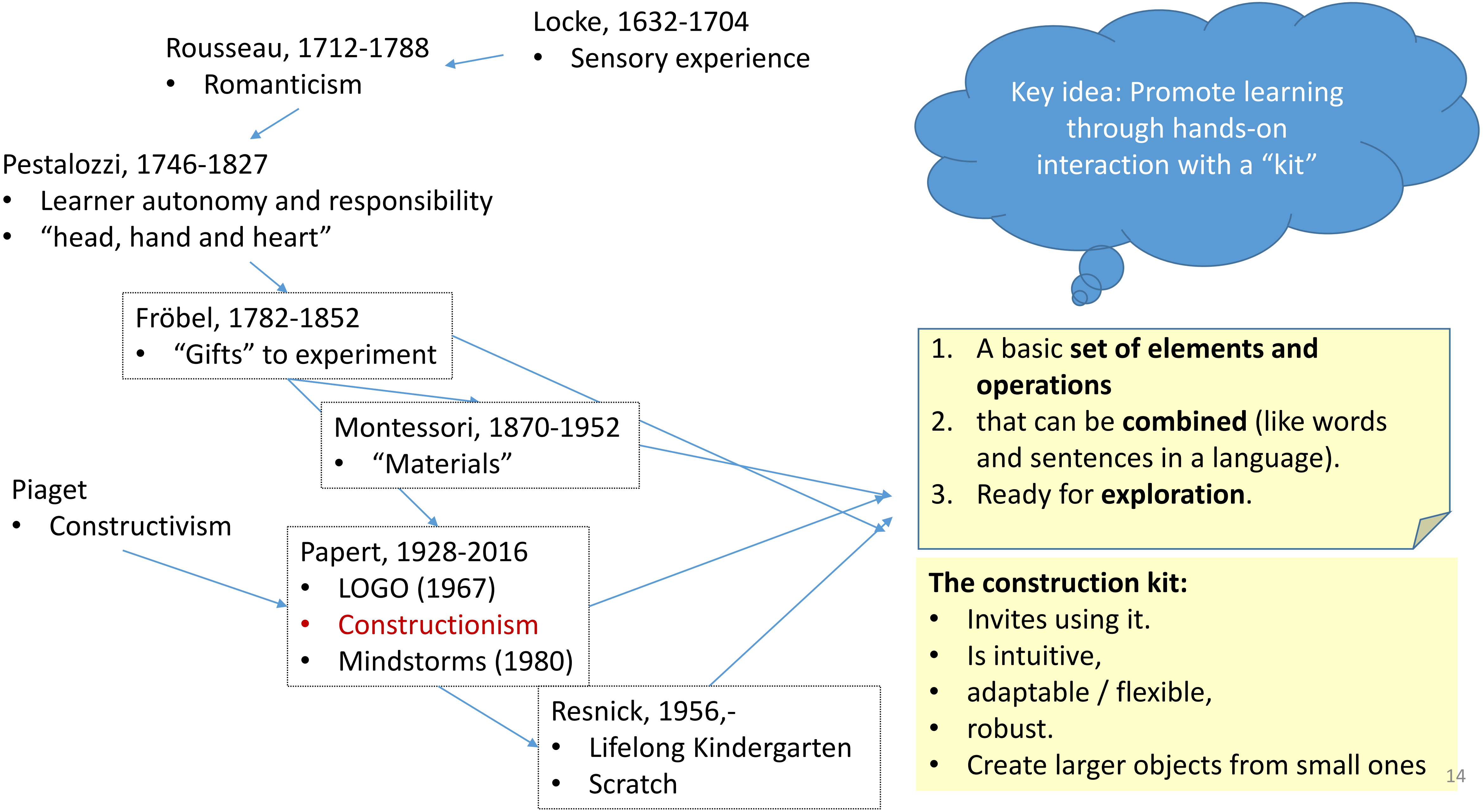








1. Learning by manipulating and constructing



Locke, 1632-1704
- Sensory experience

Rousseau, 1712-1788
- Romanticism

Pestalozzi, 1746-1827
- Learner autonomy and responsibility
- “head, hand and heart”

Fröbel, 1782-1852
- “Gifts” to experiment

Montessori, 1870-1952
- “Materials”

Piaget
- Constructivism

Papert, 1928-2016
- LOGO (1967)
- Constructionism
- Mindstorms (1980)

Resnick, 1956,-
- Lifelong Kindergarten
- Scratch

Key idea: Promote learning through hands-on interaction with a “kit”

1. A basic **set of elements and operations**
2. that can be **combined** (like words and sentences in a language).
3. Ready for **exploration**.

The construction kit:
- Invites using it.
- Is intuitive,
- adaptable / flexible,
- robust.
- Create larger objects from small ones

2. Activity-based learning

Key idea: Learning takes place in a social, cultural and material context

Karl Marx, 1818-1883

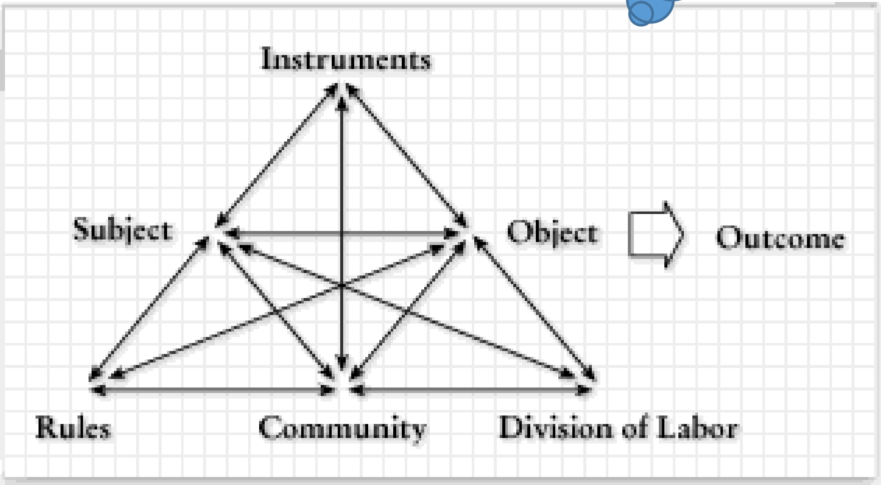
Pavlov, 1849-1936

Vygotski, 1896-1934
- Socio-constructivism
- Zone of proximal development

Leontief, 1903-1979
- Activity theory (USSR)

Activity theory (Scandinavia)
- Expansive learning (Engeström, 1987)

Nardi, 1995 (use in HCI)

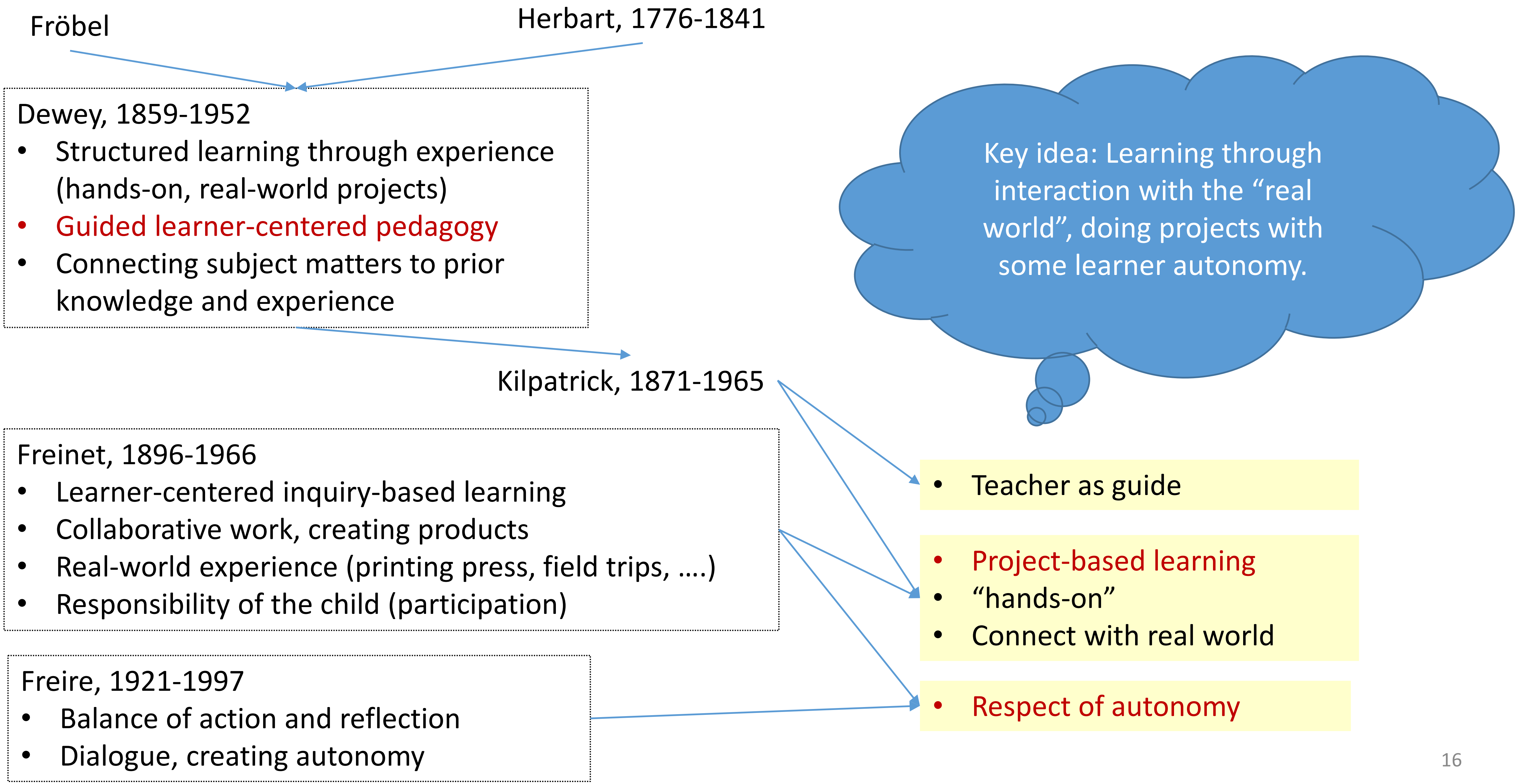


Learning happens through reflection of social knowledge.

Activities are:
- focused on objects carrying culture;
- mediated by tools carrying culture;
- Socially organized

Activities are hierarchical: activity (needs, motivation), action (goal), operation (task);

3. Hands-on, “real-world” projects



Fröbel

Herbart, 1776-1841

Dewey, 1859-1952
- Structured learning through experience (hands-on, real-world projects)
- Guided learner-centered pedagogy
- Connecting subject matters to prior knowledge and experience

Kilpatrick, 1871-1965

Freinet, 1896-1966
- Learner-centered inquiry-based learning
- Collaborative work, creating products
- Real-world experience (printing press, field trips, ….)
- Responsibility of the child (participation)

Freire, 1921-1997
- Balance of action and reflection
- Dialogue, creating autonomy

Key idea: Learning through interaction with the “real world”, doing projects with some learner autonomy.

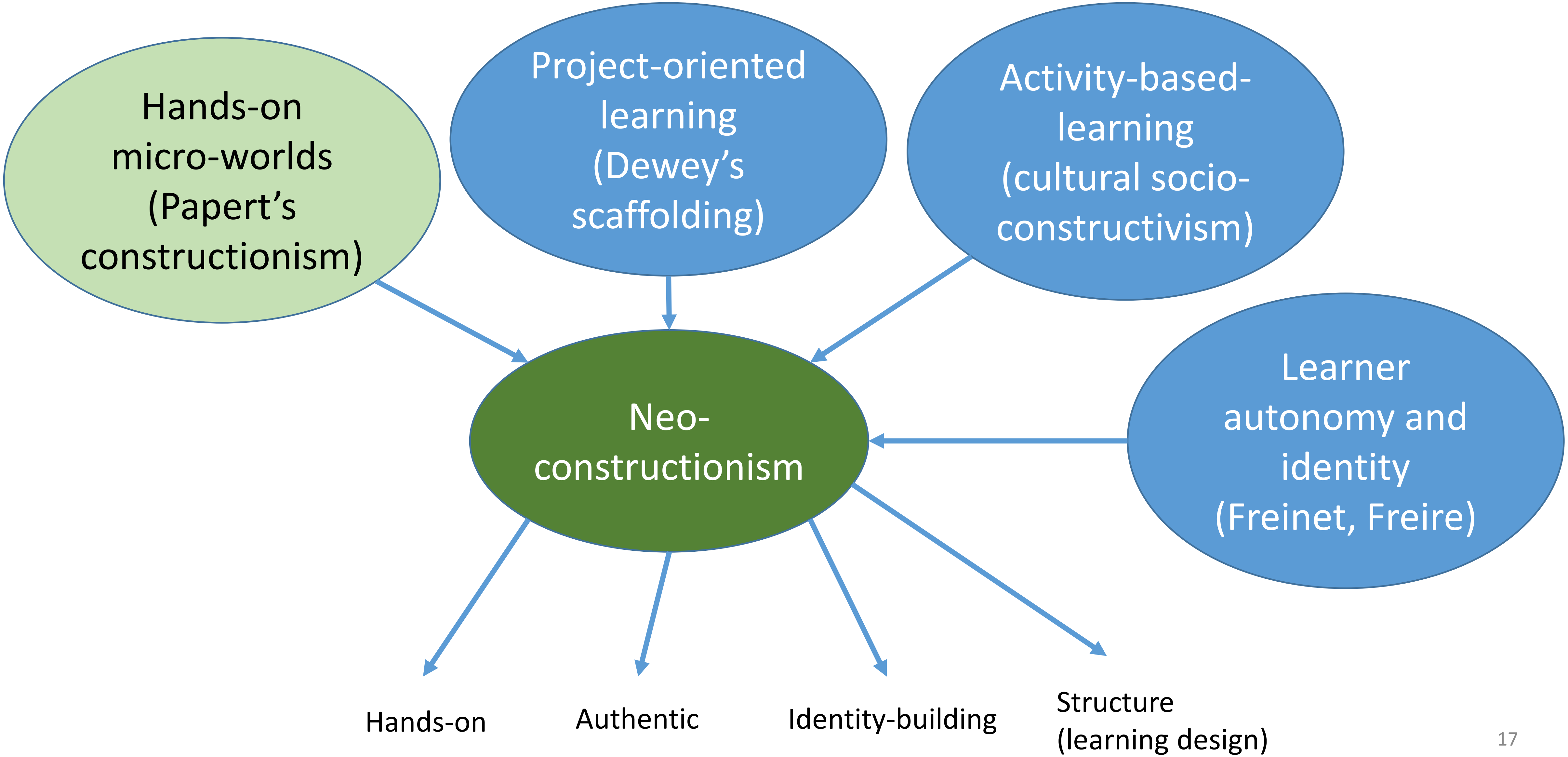
- Teacher as guide

- Project-based learning
- “hands-on”
- Connect with real world

- Respect of autonomy

4. Synthesis of theory

(found in the making & education community)



Hands-on micro-worlds (Papert's constructionism)

Project-oriented learning (Dewey's scaffolding)

Activity-based-learning (cultural socio-constructivism)

Learner autonomy and identity (Freinet, Freire)

Neo-constructionism

Hands-on

Authentic

Identity-building

Structure
(learning design)

https://tecfa.unige.ch/tecfa/talks/schneide/crs4-2020/

A short synthesis

3. COMPUTATIONAL MAKING

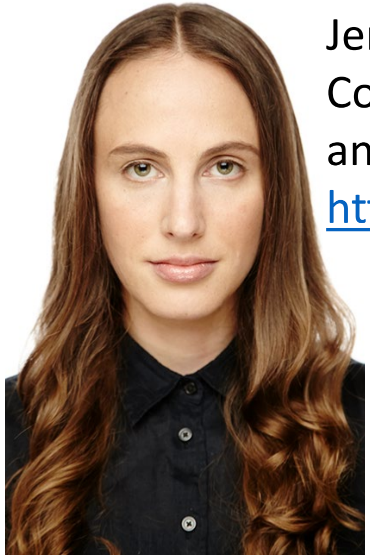
ENVIRONMENTS & THEORY

Leah Buechley,
Inventor, LilyPad
Computer science and making prof.
http://leahbuechley.com/

Limor Fried
Founder and CEO, Adafruit
https://www.adafruit.com/about/

Ayah Bdeir
Founder, LittleBits
https://ayahbdeir.com/

… and more
(including men)



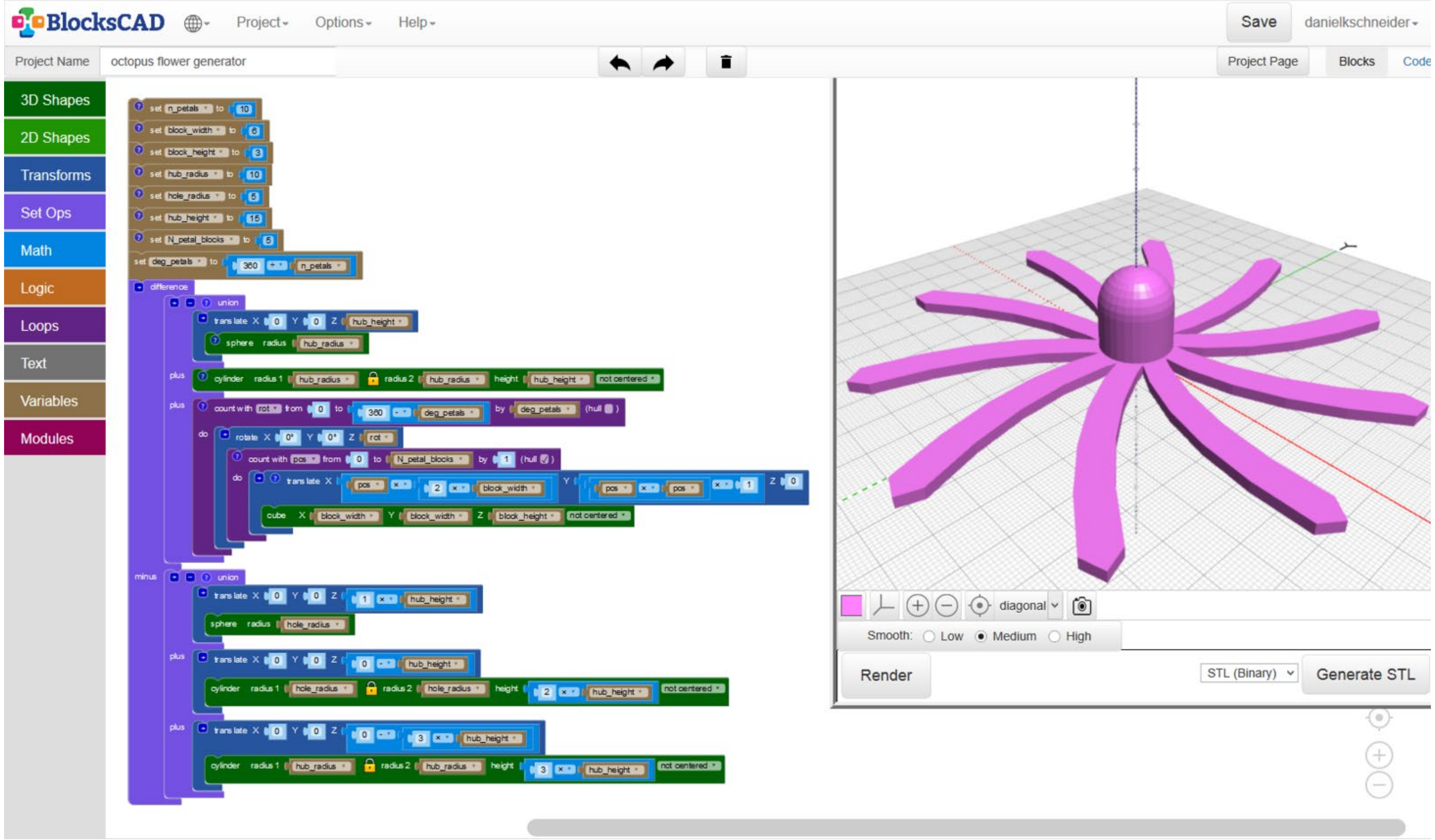
Jennifer Jacobs
Computational fashion
and art professor
http://jenniferjacobs.mat.ucsb.edu/

Kylie Peppler
Computer & education professor
Creative coding
http://kpeppler.com/

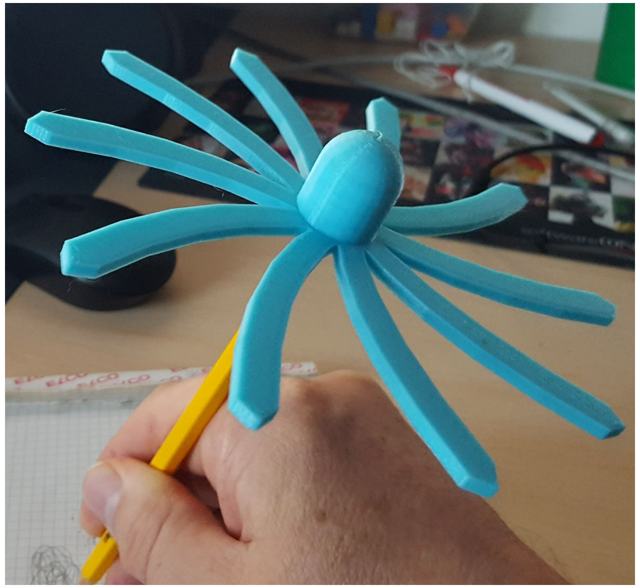
Eva S. Katterfeldt
Computer science
education + making
Researcher
http://dimeb.informatik.uni-bremen.de/

Programmed 3D objects - https://www.blockscad3d.com/

Computational making environment example



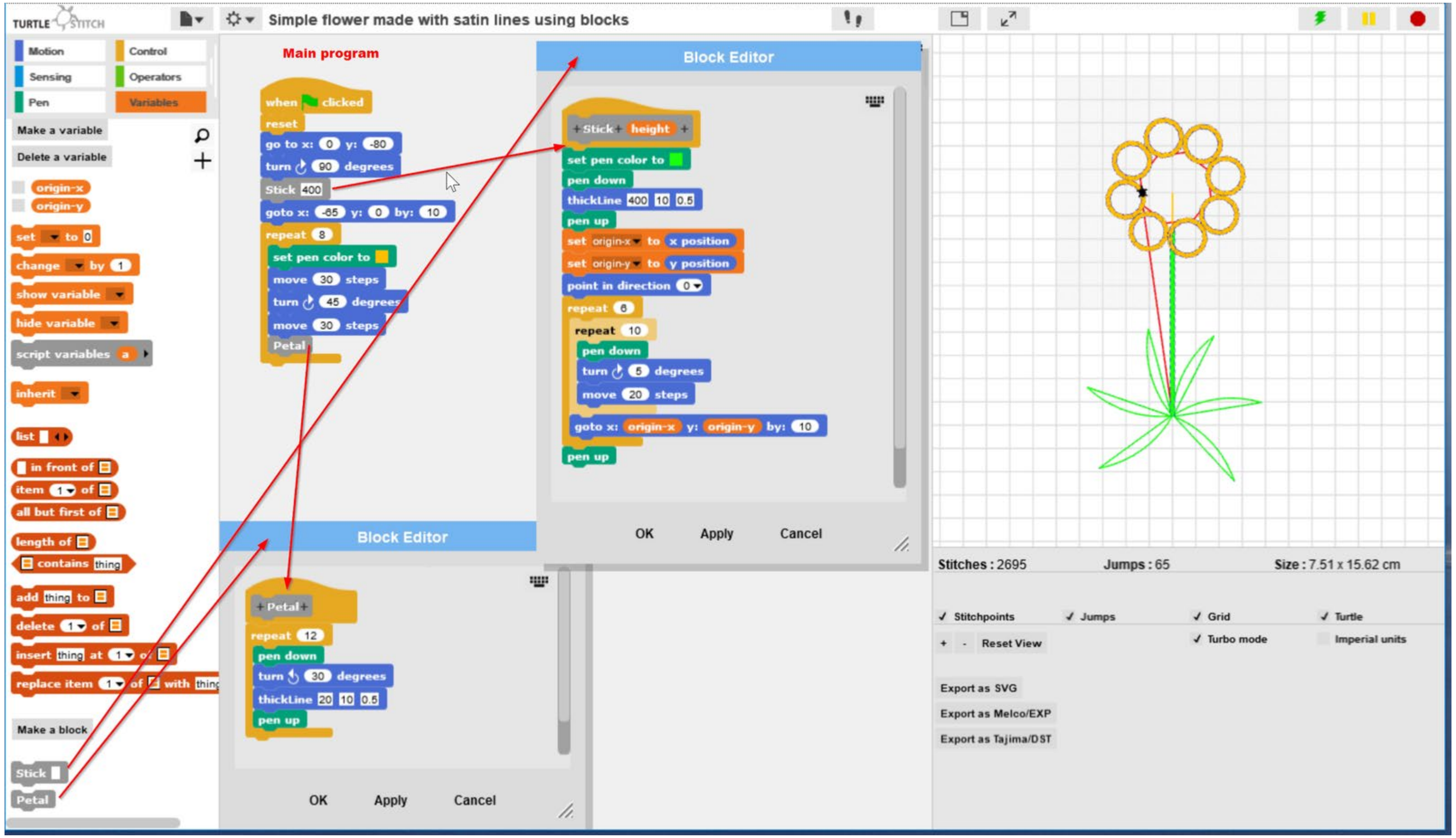
```
set n_petals to 10
set block_width to 6
set block_height to 3
set hub_radius to 10
set hole_radius to 5
set hub_height to 15
set N_petal_blocks to 5
set deg_petals to 360 ÷ n_petals
difference
        union
            translate X 0 Y 0 Z hub_height
              sphere radius hub_radius
      plus  cylinder radius 1 hub_radius radius 2 hub_radius height hub_height not centered
      plus  count with rot from 0 to 360 - deg_petals by deg_petals (hull ☐)
        do  rotate X 0° Y 0° Z rot
              count with pos from 0 to N_petal_blocks by 1 (hull ☑)
              do  translate X pos × 2 × block_width Y pos × pos × 1 Z 0
                    cube X block_width Y block_width Z block_height not centered
minus   union
            translate X 0 Y 0 Z 1 × hub_height
              sphere radius hole_radius
      plus  translate X 0 Y 0 Z 0 - hub_height
              cylinder radius 1 hole_radius radius 2 hole_radius height 2 × hub_height not centered
      plus  translate X 0 Y 0 Z 0 - 3 × hub_height
              cylinder radius 1 hub_radius radius 2 hub_radius height 3 × hub_height not centered
```



See also: https://www.tinkercad.com/learn/codeblocks

Coded embroidery - https://www.turtlestitch.org/

Computational making environment example



Simple flower made with satin lines using blocks

Main program

```
when ⚑ clicked
reset
go to x: 0 y: -80
turn ↻ 90 degrees
Stick 400
goto x: -65 y: 0 by: 10
repeat 8
  set pen color to ■
  move 30 steps
  turn ↻ 45 degrees
  move 30 steps
  Petal
```

Block Editor

```
+ Stick + height +
set pen color to ■
pen down
thickLine 400 10 0.5
pen up
set origin-x to x position
set origin-y to y position
point in direction 0
repeat 6
  repeat 10
    pen down
    turn ↻ 5 degrees
    move 20 steps
  goto x: origin-x y: origin-y by: 10
pen up
```

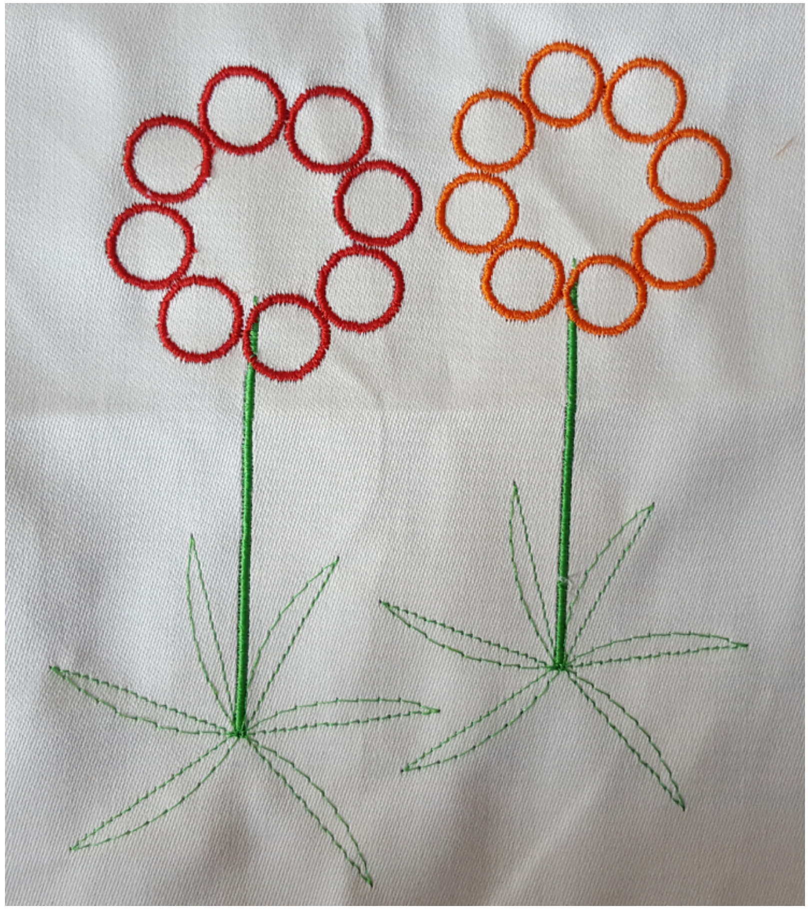
Block Editor

```
+ Petal +
repeat 12
  pen down
  turn ↺ 30 degrees
  thickLine 20 10 0.5
  pen up
```

Stitches : 2695 Jumps : 65 Size : 7.51 x 15.62 cm

✓ Stitchpoints ✓ Jumps ✓ Grid ✓ Turtle

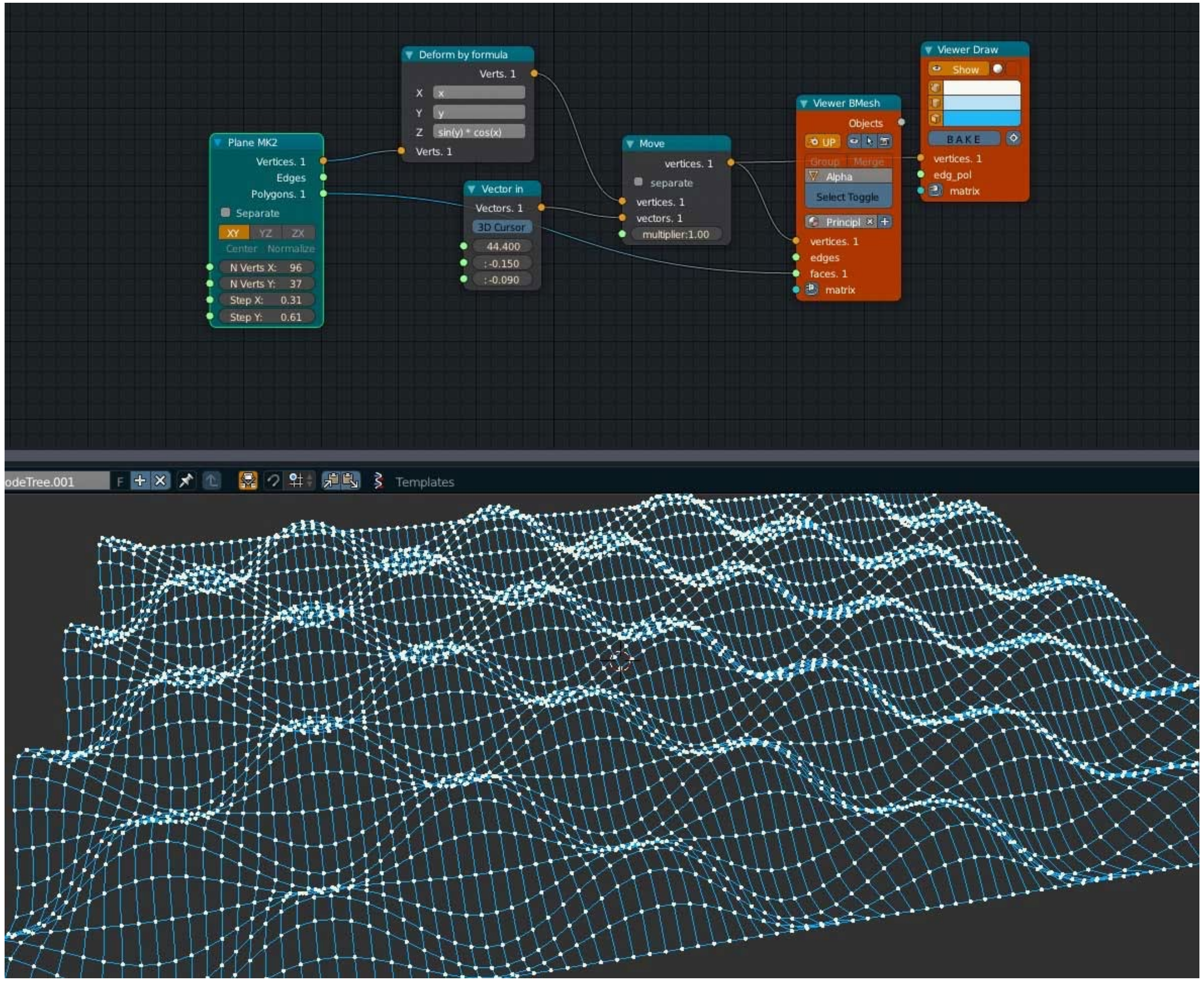
✓ Turbo mode □ Imperial units



«Nodes» programming

http://nikitron.cc.ua/sverchok_en.html

Computational design environment example



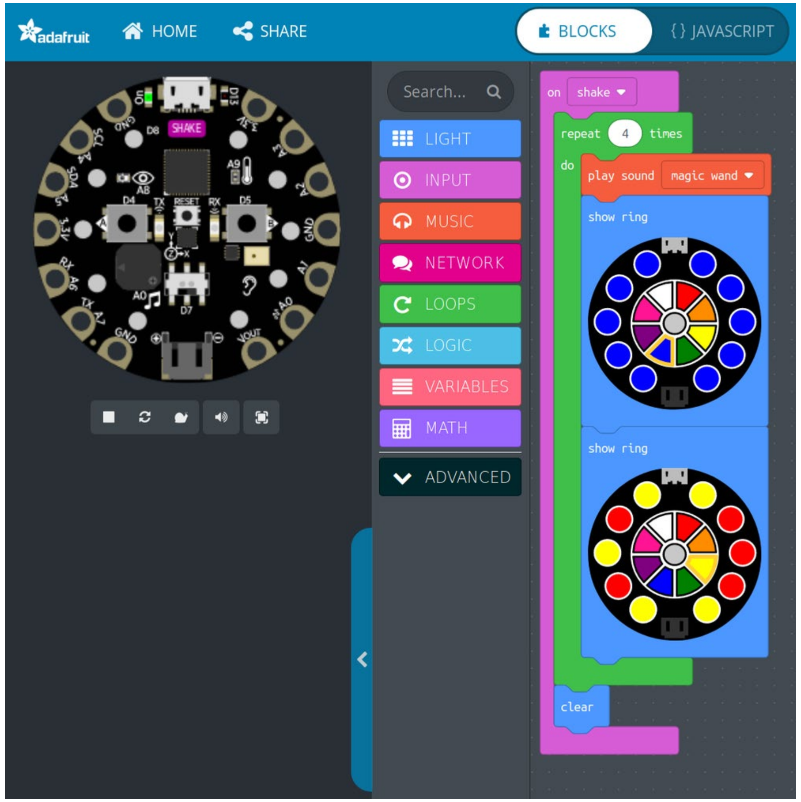
Sverchok is a Blender extension that implements over 50 nodes. See also: Grasshopper, a Rhino extension

Image: https://poneill.co/thelab/%D1%81%D0%B2%D0%B5%D1%80%D1%87%D0%BE%D0%BA/

Digital Electronics – with Adafruit CPX

https://makecode.adafruit.com/

Computational making environment example

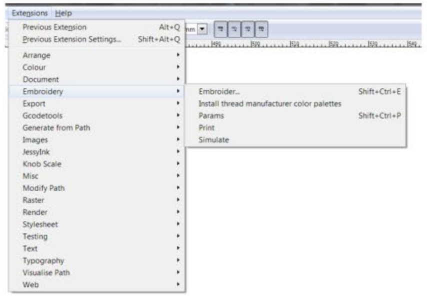


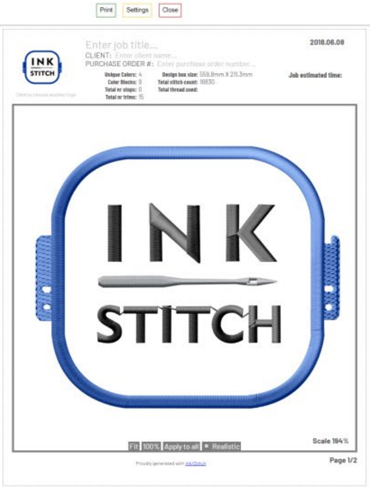
```
on shake
  repeat 4 times
  do
    play sound magic wand
    show ring
    show ring
  clear
```

Using the InkScape opensource drawing program + and its Ink/Stitch embroidery extension

Understanding file systems, drawing and parameterization

https://inkstitch.org/





Enter job title...
CLIENT: Enter client name...
PURCHASE ORDER #: Enter purchase order number...

2018.06.08

Unique Colors: 4
Color Blocks: 9
Total nr stops: 0
Total nr trims: 15

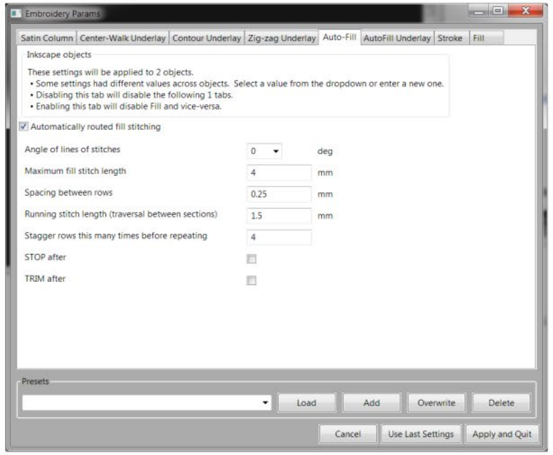
Design box size: 558.8mm X 211.3mm
Total stitch count: 18830
Total thread used:

Job estimated time:

Scale 194%

Proudly generated with Ink/Stitch

Page 1/2



Inkscape objects

These settings will be applied to 2 objects.
- Some settings had different values across objects. Select a value from the dropdown or enter a new one.
- Disabling this tab will disable the following 1 tabs.
- Enabling this tab will disable Fill and vice-versa.

☑ Automatically routed fill stitching

Angle of lines of stitches 0 deg
Maximum fill stitch length 4 mm
Spacing between rows 0.25 mm
Running stitch length (traversal between sections) 1.5 mm
Stagger rows this many times before repeating 4
STOP after ☐
TRIM after ☐

inkstitch.org

Features of digital making environments

Programming (computational making)

1. Visual languages include standard elements (Chitas et al, 2018),
2. export to symbolic languages.
3. Support for sharing code.
4. Direct creation of machine-usable formats.

Other ICT

1. System administration
2. Technical drawing
3. Image manipulation
4. Parameterization

Constructionist thoughts:

Learning is favored:

- **by manipulation and discovery**
- **by providing structure to activities**
- **by inviting dialogue**

So far, most research is theory building and UX testing (an initial search produced about 50 publications)

Feelings of engagement and empowerment fostered by these experiences indicate that computational-design tools for novices could serve as a powerful way to positively change people's understanding of the relevance and applications of programing, while fostering technological and aesthetic literacy in the process (Jacobs and Buechely, 2013)

The data shows that building the projects in our structured curriculum impacts builders' technological self-efficacy, leading to in an increase in students' comfort with, enjoyment of, and interest in programming and electronics. (Qiu et al. 2013)

motivation

Self-efficacy

usefulness

«Bildung»

The creation of computational artefacts as a means of expression could be an exciting way to develop computational literacy. (Chytax, Tsilingiris and Diethelm, 2019)

In addition to constructionism, the interaction between body and mind, creativity and technology and self and environment.", i.e. *be-greifbarkeit, imagineering* and *self-efficacy* are essential requirements for learning environments for digital fabrication that facilitate *Bildung (*Katterfeldt and Dittert, 2015*)*

A school may purchase a 3D printer for educational purposes, only to have its student-makers print other people's models without learning to make their own. To prevent this kind of situation, educators must capitalize on the maker movement in ways that facilitate what we call computational making, which involves both meaningful cognition and the making of artifacts. (Johnson, 2017)

For the references, see: https://edutechwiki.unige.ch/en/Computational_making

4. TEACHER-CREATED LEARNING OBJECTS

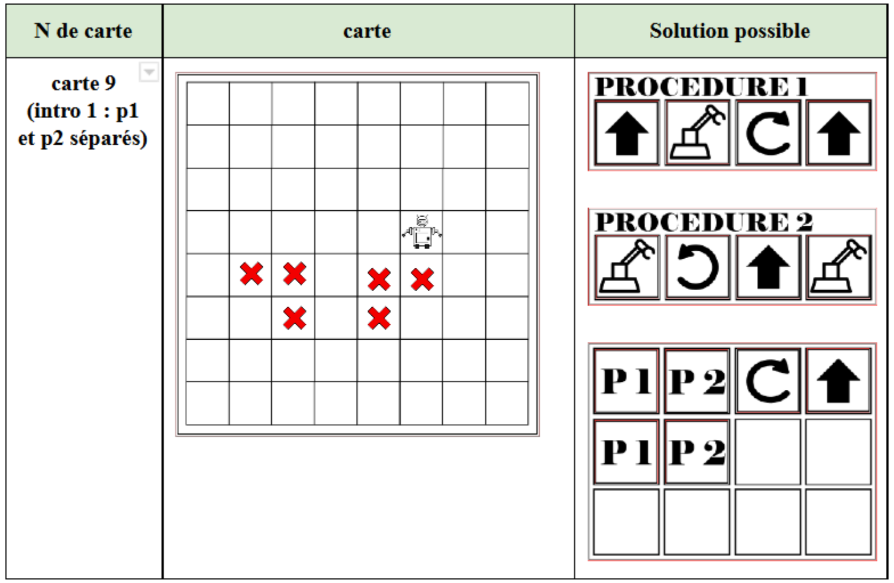
Why «teacher making»?

1. Teachers who teach «making» must be trained or learn it themselves.
   Creating objects for one's own teaching is a good & motivating way to start "making" and in line with our neo-constructivist principles

2. To create educational objects that should should trigger cognitive and social processes that are good for learning

There is little research on the subject (Boufflers & Schneider, submitted)

Teach programming principles with unplugged programming



N de carte	carte	Solution possible
carte 9 (intro 1 : p1 et p2 séparés)		PROCEDURE 1 PROCEDURE 2 P 1 P 2 P 1 P 2

Plateau de jeu

Boty, le robot

map

Support Principal

Supports procédures

Programming supports

Pièces de déplacement

Outils

Boite à outils

instructions

Example:

- Programming Boty
- Made with a laser cutter

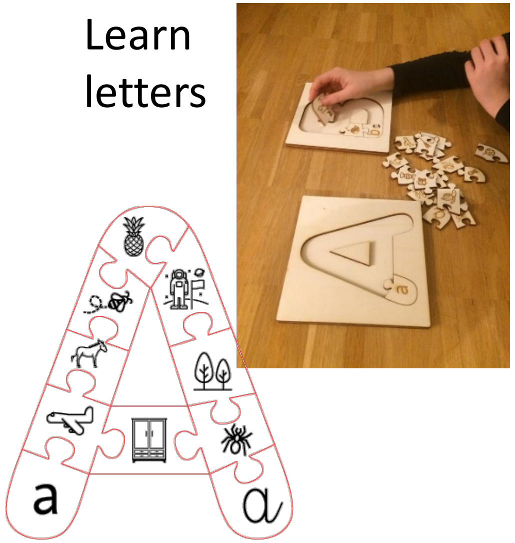
Ojectives

- Manipulate to learn abstract concepts: instructions and <u>procedures /calls</u>

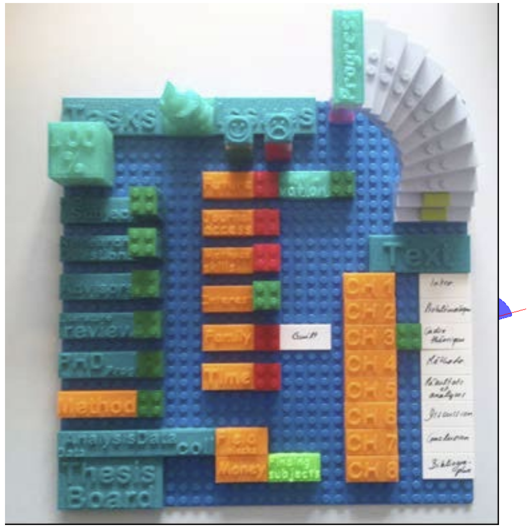
… and create/use many more kinds of teaching & learning tools

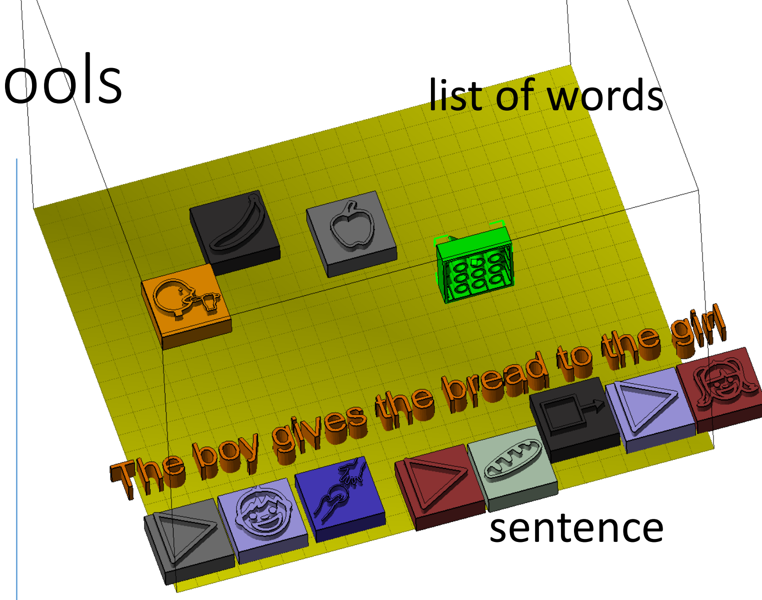
Learn about map topology

Learn letters



Project management with Lego and compatibles





Communicate with children that have cognitive trouble

Large molecules and small drugs

Cox proteine

Drug : "Flubiprofen"

Drug : "Celecoxib" (non nettoyé)

Drug : "Diclofenac"

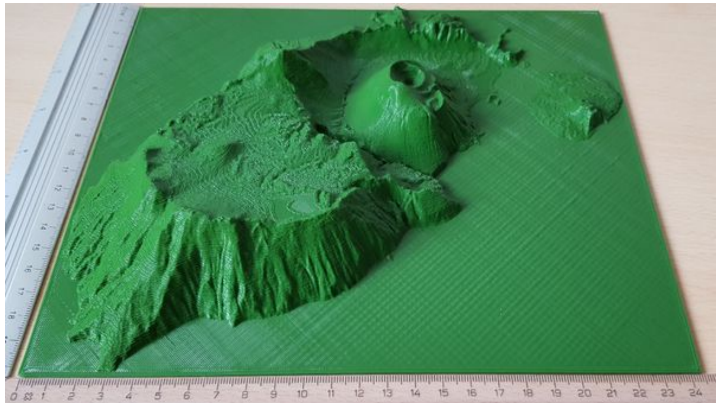
Drug : "Nimesulid"

Drug : "Salicylic"

Drug : "Ibuprofen" (non nettoyé)

Drugs that fit

Isola di Vulcano, risk management



Tokens for classrooms

See also: http://tecfaetu.unige.ch/digifabwiki/index.php/Liste_des_projets

Early prototyping: engagement in thinking and collaboration *with* “common” manipulables can help shaping ideas, concepts and projects

90 min. workshop on defining roles for team work (including prototyping role tokens), July 12, 2017, SDG Summer School, UniGE/Tsinghua

5. CONCLUSION AND OUTLOOK

Digital design and manufacturing has the potential to improve pedagogies - including computational thinking - through (improved) authenticity, graspability, self-efficacy and structure

Some working hypothesis for further work:

- Computational making environments allow teaching key components of computational thinking
- The overhead of «making» is counterbalanced by other benefits, including motivation and self-efficacy.
- Making favors integration of knowledge and authenticity
- Making could develop 21st century soft-skills like design thinking, collaboration, and sharing
- Digital manufacturing allow teachers to create objects that are useful (effective) and usable.
- Design of physical tools leads to teachers to think educational activities in a different way.
- Computational making can attract different populations to ICT

Thank you for listening

- Questions ?
- Comments ?
- Your own work / experience ?

Slides:

https://tecfa.unige.ch/tecfa/talks/schneide/crs4-2020/

My preparation materials in our wiki:

https://edutechwiki.unige.ch/en/Computational_making
https://edutechwiki.unige.ch/fr/CFAO